CS 14 Spring 2001 — Mid-term #2

`Name: _____`

You have 50 minutes for this exam, so use your time wisely. **Read carefully** and **answer clearly** each question! You will find, attached to the back of your exam, a **copy of the single-cycle and multicycle datapaths** for your reference.

1. (30 points) The following questions (on two pages) require **only short answers:**
   - Show the subtraction `13` from `9` using **5-bit, twos complement numbers**. Show that your result is, in fact, `-4`.

   - Why must the control unit for for a multicycle processor be **sequential**, and not **combinational**?

- Give an example of a **control hazard** for a pipelined processor, and indicate one way in which that control hazard can handled correctly.

- To perform subtraction, the ALU negates the `B` input and and then adds the result to the `A` input. How does an ALU perform the negatiion of the `B` input?

2. (35 points) Consider a new type of instruction that we will call a **CAS-type** (*compute-and-store type*). These instructions look very much like R-type instructions, but are interpreted differently. The following is an example of a CAS-type instruction:

```
addcas $s0, $t0, $t1
```

This instruction will sum the contents of registers `$t0` and `$t1`, and store the result *in main memory at the address specified in register* `$s0`. More generally put, for CAS-type instructions, the destination register is not itself the place in which the result will be stored, but rather contains the address in main memory at which the result will be stored.

Construct a complete datapath for a single cycle processor that supports not only the R-type, lw/sw, and branch instructions, but also CAS-type instructions. That is, build upon the single cycle datapath provided to support CAS-type instructions.

3. (35 points) Consider another new instruction that *swaps the contents of two registers*:

   ```
   swap $s3, $s4
   ```

   The two registers are encoded as the `rs` and `rt` components of the instruction. **Enumerate the steps for multicycle processor needed to execute this instruction.** Describe clearly each step in terms of the components shown on the included multicycle datapath. Remember that for all instructions, the first step is the *instruction fetch* step, and the second step reads from the register bank the two source registers that may be specified by the instruction.